

*File Copy*

BRL MR 2060

# BRL

AD 712946

MEMORANDUM REPORT NO. 2060

SYMAP1 - AN EXPERIMENTAL SYMBOL MANIPULATION PROGRAM

by

George C. Francis

August 1970

This document has been approved for public release and sale;  
its distribution is unlimited.

U.S. ARMY ABERDEEN RESEARCH AND DEVELOPMENT CENTER  
**BALLISTIC RESEARCH LABORATORIES**  
ABERDEEN PROVING GROUND, MARYLAND

Destroy this report when it is no longer needed.  
Do not return it to the originator.

The findings in this report are not to be construed as  
an official Department of the Army position, unless  
so designated by other authorized documents.

BALLISTIC RESEARCH LABORATORIES

MEMORANDUM REPORT NO. 2060

AUGUST 1970

SYMAP1 - AN EXPERIMENTAL SYMBOL MANIPULATION PROGRAM

George C. Francis

Applied Mathematics Division

This document has been approved for public release and sale;  
its distribution is unlimited.

RDT&E Project No. 1T061102A14B

ABERDEEN PROVING GROUND, MARYLAND

BALLISTIC RESEARCH LABORATORIES

MEMORANDUM REPORT NO. 2060

GCFrancis/bj  
Aberdeen Proving Ground, Md.  
August 1970

SYMAP1 - AN EXPERIMENTAL SYMBOL MANIPULATION PROGRAM

ABSTRACT

SYMAP1 is a BRLESC computer program designed to carry out a variety of algebraic symbol manipulations including arithmetic operations, substitution, certain kinds of simplifications, and rudimentary differentiation. Main emphasis has been on polynomials in several variables (including truncated power series), but certain other mathematical forms can also be manipulated. Examples of certain applications to numerical analysis and theory of equations are included.

# TABLE OF CONTENTS

	Page
ABSTRACT. . . . .	3
I. INTRODUCTION TO SYMAP1. . . . .	7
II. EXAMPLE FROM NUMERICAL ANALYSIS . . . . .	10
III. INTERNAL NOTATION FOR CONSTANTS, VARIABLES, AND FUNCTION NAMES. . . . .	17
IV. EXAMPLE FROM THE THEORY OF CUBIC EQUATIONS. . . . .	18
V. SIMPLIFICATION OF FORMS . . . . .	24
VI. DETAILS OF SPECIFICATION OF MANIPULATIONS . . . . .	25
VII. EVALUATION OF EXPRESSIONS . . . . .	27
VIII. VECTOR OPERATIONS . . . . .	29
IX. CONCLUSION. . . . .	30
REFERENCES. . . . .	31
DISTRIBUTION LIST . . . . .	33

## I. INTRODUCTION TO SYMAP1

In order to relieve a researcher of certain straightforward but tedious algebraic symbol manipulations, a BRLESC 1 computer program called SYMAP1 has been developed by the author. Sufficient capability has been demonstrated to permit announcing its availability. Additional features will be added as they are found useful, and more powerful techniques are being studied.

Major emphasis in the early stages has been on manipulation of polynomials in several variables (including truncated power series), but many features of SYMAP1 are not restricted to polynomials. Principal manipulations of expressions include the following (see also Figure 1):

Manipulations:	Symbolic Commands:
1. Indicated sum of two expressions	ADD
2. Indicated difference	SUBTR
3. Indicated product	MULT
4. Indicated quotient	DIV
5. Indicated power	EXPON
6. Certain substitutions in expressions	
a. Name for name	SUBSTN
b. Name for expression	SUBSTNE
c. Expression for name	SUBSTE
d. Expression for expression	SUBSTEE
7. Display of any previously stored labeled expression	PRINT

In the above cases the expressions have few restrictions. In order that simplifications of the results may be virtually automatic, however, some limitations have to be imposed at present. Since the user may want to retain the unsimplified results for further use, simplification is considered a separate manipulation:

- |                              |         |
|------------------------------|---------|
| 8. Simplification (limited)  | SIMEXPR |
| 9. Differentiation (limited) | DIFF    |

At present differentiation is limited to polynomial - like expressions in the variable of differentiation (negative and decimal fraction exponents are allowed). Products and certain more general expressions can be handled by considering the parts separately and combining in several steps.

Manipulations:					Descriptions:
ADD SIMEXPR	PA PC1	PB PC	PC1		Add PA and PB to obtain PC1 and simplify as PC
SUBTR SIMEXPR	PA PD1	PB PD	PD1		Subtract PB from PA to obtain PD1 and simplify as PD
MULT SIMEXPR	PT PE1	PA PE	PE1		Multiply PT times PA to obtain PE1 and simplify as PE
DIV SIMEXPR	PB PF1	PT PF	PF1		Divide PB by PT to obtain PF1 and simplify as PF
EXPON SIMEXPR	PT PG1	PK PG	PG1		Exponentiate PT to PK to obtain PG1 and simplify as PG
SUBSTEE SIMEXPR	PT PH1	X PH	PA	PH1	Substitute expression PT for expression X in PA to obtain PH1 and simplify as PH
SUBSTEE SIMEXPR	PT QB1	Z QB	QA	QB1	Substitute expression PT for expression Z in QA to obtain QB1 and simplify as QB
DIFF	R1	X	DR1		Differentiate R1 w.r.t. X to obtain DR1

Figure 1. Sample Manipulations  
(Refer also to Figure 2)

Given:

PA	$1.3*X^{**2}-.5*X^{**3}+.6*Y^{**1.7}$
PB	$X^{**}-1.+1.3*X^{**2}-.7*Y^{**1.7}$
PT	$2.*Y^{**3}.$
PK	$3.$
X	X
Y	Y
Z	Z1+Z2
QA	$(Z1+Z2)^{**2}/Y+\cos(XI)+Z1+Z2$
R1	$3.*X^{**2}.*Y+X^{**}-1./(Y+Z)$
,	,

Results (which can be manipulated further):

PC1	$1.3*X^{**2}-.5*X^{**3}+.6*Y^{**1.7}+X^{**}-1.+1.3*X^{**2}-.7*Y^{**1.7}$
PC	$2.6*X^{**2}-.5*X^{**3}+X^{**}-1.+5.3*Y^{**1.7}$
PD1	$1.3*X^{**2}-.5*X^{**3}+.6*Y^{**1.7}-(X^{**}-1.+1.3*X^{**2}-.7*Y^{**1.7})$
PD	$-.5*X^{**3}.-X^{**}-1.+6.7*Y^{**1.7}$
PE1	$2.*Y^{**3}.*(1.3*X^{**2}-.5*X^{**3}+.6*Y^{**1.7})$
PE	$2.6*X^{**2}.*Y^{**3}.-X^{**3}.*Y^{**3}+.12.*Y^{**4.7}$
PF1	$(X^{**}-1.+1.3*X^{**2}-.7*Y^{**1.7})/(2.*Y^{**3}.)$
PF	$.5*X^{**}-1.*Y^{**}-3.+65*X^{**2}.*Y^{**}-3.-.35*Y^{**}-1.3$
PG1	$(2.*Y^{**3}.)^{**3}.$
PG	$8.*Y^{**9}.$
PH1	$1.3*(2.*Y^{**3}.)^{**2}-.5*(2.*Y^{**3}.)^{**3}+.6*Y^{**1.7}$
PH	$6.*Y^{**1.7}+5.2*Y^{**6}.-4.*Y^{**9}.$
QB1	$(2.*Y^{**3}.)^{**2}/Y+\cos(XI)+2.*Y^{**3}.$
QB	$2.*Y^{**3}+.4.*Y^{**5}+\cos(XI)$
DR1	$6.*X*Y-X^{**}-2.*(Y+Z)^{**}-1.$

Figure 2. Sample Input and Results  
(Refer also to Figure 1)

Certain difficulties are circumvented by doing some of the manipulations on algebraic forms expressed in a non-standard notation (called here canonical notation). Provision is made to translate (simple) expressions to and from this canonical notation and to manipulate expressions which are in canonical form:

10. Convert to canonical form	CONVERT
11. Reconvert from canonical form	RECONVERT
12. Add in canonical form	CADD
13. Subtract	CSUBT
14. Multiply	CMULT
15. Divide	CDIV
16. Exponentiate	CEXP
17. Simplify	CSIME
18. Differentiate	CDIFF
19. Substitute expression for expression	CSBEE
20. Print, one term per line	SPREAD

The last (SPREAD), usable in either canonical or regular notation, is sometimes more convenient than PRINT when expressions involve many terms in several variables.

A realistic practical example will be considered next, using canonical forms largely.

## II. EXAMPLE FROM NUMERICAL ANALYSIS

In designing numerical approximations for partial derivatives for use in solving partial differential equations, the following expression can be considered in parabolic cases:

$$G(\xi, \eta) = (Ae^{-\xi} + B + Ce^{\xi})e^{\frac{1}{2}\eta} + (De^{-\xi} + E + Fe^{\xi})e^{-\frac{1}{2}\eta}$$

Here A, B, C, D, E, F are undetermined weight factors to apply to function values at points of a rectangular grid neighboring to the point at which the derivatives are evaluated.  $\xi$  and  $\eta$  are related to step-sizes in two orthogonal directions (grid distances).

It is often desirable to relate the step-sizes by setting

$$\eta = P\xi^2$$

where P is another undetermined constant.

Useful relations between the weights A,...,F can be obtained by expanding  $G(\xi, \eta)$  in powers of  $\xi$  and  $\eta$  and then replacing  $\eta$  or by expanding instead  $H(\xi) = G(\xi, P\xi^2)$  in powers of  $\xi$ , and setting the coefficient of each low power of  $\xi$  equal to zero.

The currently available program SYMAP1 can assist in this effort. If the exponentials are replaced by truncated power series,  $G(\xi, \eta)$  becomes an expression involving polynomials in  $\xi$  and  $\eta$ , and  $H(\xi)$  likewise in  $\xi$  alone. The multiplying, adding, substituting (in some cases), rearranging, and combining of like terms can be done by SYMAP1 on the computer BRLESC 1.

Figure 3 shows a possible sequence of SYMAP1 steps to achieve the result. Truncation of all series through a given power of  $\xi$  is arranged by specifying a parameter and setting a program option. Terms through power 8 can be handled in currently available memory. Note that  $G(\xi, \eta)$  would involve several dozen terms in this case, a situation in which hand computation might easily involve careless errors. It was to avoid such errors that SYMAP1 was originally proposed.

In Figure 3 the initial input (from cards) is a sequence of labeled expressions which are to be operated on. The end of this set of lines is indicated by a special sentinel card with commas in card columns 1 and 2. The labels are arbitrary (up to nine characters) but here are shown as rough mnemonics in some cases, such as EXI for  $e^\xi$ , EMHETA for  $e^{-\frac{1}{2}\eta}$ , EHETX for  $e^{\frac{1}{2}\eta}$  in terms of  $\xi$ , etc. The power series are truncated here after  $\xi^4$  or, equivalently,  $\eta^2$ .

Following the sentinel card are suitable SYMAP1 instructions, one per card. Many alternative sequences could have been used, but here the labeled expressions wanted are all converted into a convenient internal notation called canonical notation prior to any combining.

Input:

EXI  $1.+XI^{**1}+.0.5*XI^{**2}+.16666667*XI^{**3}+.04166667*XI^{**4}.$   
 EMXI  $1.-XI^{**1}+.0.5*XI^{**2}-.16666667*XI^{**3}+.04166667*XI^{**4}.$

EHETA  $1.+0.5*ETA^{**1}+.0.125*ETA^{**2}.$   
 EMHETA  $1.-0.5*ETA^{**1}+.0.125*ETA^{**2}.$

A  $A^{**1}.$   
 B  $B^{**1}.$   
 C  $C^{**1}.$   
 D  $D^{**1}.$   
 E  $E^{**1}.$   
 F  $F^{**1}.$

EHETX  $1.+0.5*P^{**1}*XI^{**2}+.0.125*P^{**2}*XI^{**4}.$   
 EMHETX  $1.-0.5*P^{**1}*XI^{**2}+.0.125*P^{**2}*XI^{**4}.$

,,

Steps:

CONVERT	A	CA		
CONVERT	B	CB		
CONVERT	C	CC		
CONVERT	D	CD		
CONVERT	E	CE		
CONVERT	F	CF		
CONVERT	EXI	CEXI		$e^{\xi}$
CONVERT	EMXI	CEMXI		$e^{-\xi}$
CONVERT	EHETA	CEHETA	(*)	$e^{\frac{1}{2}\eta}$
CONVERT	EMHETA	CEMHETA	(*)	$e^{-\frac{1}{2}\eta}$
CONVERT	EHETX	CEH1		$e^{\frac{1}{2}P\xi^2}$
CONVERT	EMHETX	CEMH1		$e^{-\frac{1}{2}P\xi^2}$
CMULT	CA	CEMXI	T1	$A.e^{-\xi}$
CMULT	CC	CEXI	T2	$C.e^{\xi}$
CADD	T1	CB	T3	$Ae^{-\xi} + B$
CADD	T3	T2	T4	$Ae^{-\xi} + B + Ce^{\xi}$
CMULT	T4	CEHETA	S1	(*) $(Ae^{-\xi} + B + Ce^{\xi})e^{\frac{1}{2}\eta}$
CMULT	CD	CEMXI	T11	$D.e^{-\xi}$
CMULT	CF	CEXI	T12	$F.e^{\xi}$
CADD	T11	CE	T13	$De^{-\xi} + E$
CADD	T13	T12	T14	$De^{-\xi} + E + F.e^{\xi}$
CMULT	T14	CEMHETA	S2	(*) $(De^{-\xi} + E + F.e^{\xi})e^{-\frac{1}{2}\eta}$
CADD	S1	S2	S3	(*) $G(\xi, \eta)$
RECONVERT	S3	GFCN		(*) $G(\xi, \eta)$
SPREAD	GFCN			(*) $G(\xi, \eta)$
CMULT	T4	CEH1	S11	
CMULT	T14	CEMH1	S12	
CADD	S11	S12	S13	$H(\xi) = G(\xi, P\xi^2)$
RECONVERT	S13	HFCN		
SPREAD	HFCN			
,,				(*) only if G itself is wanted

Figure 3. Input and Manipulations for First Example

```

A
+B
+C
+D
+E
+F
-XI*A
+XI*C
-XI*D
+XI*F
+.5*XI**2.*P*A
+.5*XI**2.*P*B
+.5*XI**2.*P*C
-.5*XI**2.*P*D
-.5*XI**2.*P*E
-.5*XI**2.*P*F
+.5*XI**2.*A
+.5*XI**2.*C
+.5*XI**2.*D
+.5*XI**2.*F
-.5*XI**3.*P*A
+.5*XI**3.*P*C
+.5*XI**3.*P*D
-.5*XI**3.*P*F
-.1666667*XI**3.*A
+.1666667*XI**3.*C
-.1666667*XI**3.*D
+.1666667*XI**3.*F
+.25*XI**4.*P*A
+.25*XI**4.*P*C
-.25*XI**4.*P*D
-.25*XI**4.*P*F
+.125*XI**4.*P**2.*A
+.125*XI**4.*P**2.*B
+.125*XI**4.*P**2.*C
+.125*XI**4.*P**2.*D
+.125*XI**4.*P**2.*E
+.125*XI**4.*P**2.*F
+.0416667*XI**4.*A
+.0416667*XI**4.*C
+.0416667*XI**4.*D
+.0416667*XI**4.*F

```

Figure 4.  $H(\xi)$  in "Spread" Format

The comments at the right indicate the result obtained at each step. (CADD, CSUBT, and CMULT include some automatic simplification.) When a result of sufficient interest is obtained, such as  $H(\xi)$  near the end, it can be reconverted into a more readable form and if desired printed one term per line, i.e., "spread".

In this example the final result  $H(\xi)$  is displayed in ascending powers of  $\xi$ . Terms involving the same power of  $\xi$  are thus adjacent for the convenience of the reader. (See Figure 4.)

Well-known conclusions that can be drawn from this display are that for increasingly higher accuracy of numerical approximation, the parameters  $A, \dots, F, P$  should be selected to satisfy more and more (from the top) of the following relations, one for each power of  $\xi$ :

$$\begin{aligned}
 A + B + C + D + E + F &= 0 \\
 -A + C - D + F &= 0 \\
 P(A + B + C - D - E - F) + (A + C + D + F) &= 0 \\
 3*P*(-A + C + D - F) + (-A + C - D + F) &= 0 \\
 6*P*(A + C - D - F) + 3*P^2*(A + B + C + D + E + F) + (A + C + D + F) &= 0
 \end{aligned}$$

Note that truncating after the fourth power of  $\xi$  still leads to 42 terms in the expression. This number would be much greater for 6th, 8th, etc. powers, with increasing chance of human error if done by hand. Expansions through power 8 have been run on BRLESC 1 for this problem.

In Figure 3 actual numerical coefficients (such as .125 and .16666667) were shown. After sufficient manipulation and combination such coefficients might become unrecognizable; so they could be treated as additional parameters and replaced at a later stage if desired. (See Figure 5.) This would make each term longer, however, and thus require some additional machine effort. (There is a limit on the number of factors per SYMAP1 term, currently seven in most cases, fourteen in special cases.) Figure 5 also further demonstrates that negative exponents of simple variables can be handled in SYMAP1. Decimal fractions in exponents are also allowed.

Given:

EXI1	$1. + XI^{**1} . + TWO^{**} - 1 . * XI^{**2} . + TWO^{**} - 1 . * TRE^{**} - 1 . * XI^{**3} .$
EXI2	$TWO^{**} - 3 . * TRE^{**} - 1 . * XI^{**4} .$
EMXI1	$1. - XI^{**1} . + TWO^{**} - 1 . * XI^{**2} . - TWO^{**} - 1 . * TRE^{**} - 1 . * XI^{**3} .$
EMXI2	$TWO^{**} - 3 . * TRE^{**} - 1 . * XI^{**4} .$
EHETX1	$1. + TWO^{**} - 1 . * P^{**1} . * XI^{**2} . + TWO^{**} - 3 . * P^{**2} . * XI^{**4} .$
EMHETX1	$1. - TWO^{**} - 1 . * P^{**1} . * XI^{**2} . + TWO^{**} - 3 . * P^{**2} . * XI^{**4} .$
A	$A^{**1} .$
B	$B^{**1} .$
C	$C^{**1} .$
D	$D^{**1} .$
E	$E^{**1} .$
F	$F^{**1} .$
TWO	962
TRE	963
CTWO	2.\$
CTRE	3.\$
,	,

Manipulations:

Results:

CONVERT	A	CA
CONVERT	B	CB
CONVERT	C	CC
CONVERT	D	CD
CONVERT	E	CE
CONVERT	F	CF
CONVERT	EXI1	CX1
CONVERT	EXI2	CX2
CONVERT	EMXI1	CMX1
CONVERT	EMXI2	CMX2
CONVERT	EHETX1	CH1
CONVERT	EMHETX1	CMH1

$$e^{\frac{1}{2}P\xi^2}$$

$$e^{-\frac{1}{2}P\xi^2}$$

Figure 5(a). Modified Version of First Example. Part 1

Manipulations:

Results:

CADD	CX1	CX2	CEX		$e^{\xi}$
CADD	CMX1	CMX2	CEMX		$e^{-\xi}$
CMULT	CA	CEMX	T21		$Ae^{-\xi}$
CMULT	CC	CEX	T22		$Ce^{\xi}$
CADD	T21	CB	T23		$Ae^{-\xi} + B$
CADD	T23	T22	T24		$Ae^{-\xi} + B + Ce^{\xi}$
CMULT	CD	CEMX	T31		$De^{-\xi}$
CMULT	CF	CEX	T32		$Fe^{\xi}$
CADD	T31	CE	T33		$De^{-\xi} + E$
CADD	T33	T32	T34		$De^{-\xi} + E + Fe^{\xi}$
CMULT	T24	CH1	S21		
CMULT	T34	CMH1	S22		
CADD	S21	S22	S23		$H(\xi)$
RECONVERT	S23	HXI			$H(\xi; TWO, TRE)$
SPREAD	HXI				
CSBEE	CTWO	TWO	S23	S31	Replace TWO
CSIME	S31	S32			
CSBEE	CTRE	TRE	S32	S34	Replace TRE
CSIME	S34	S35			
RECONVERT	S35	H1			$H(\xi), \text{ simplified}$
SPREAD	H1				
,,					

Figure 5(b). Modified Version of First Example. Part 2

### III. INTERNAL NOTATION FOR CONSTANTS, VARIABLES, AND FUNCTION NAMES

Variables and parameters which occur in the manipulation of algebraic expressions are represented in the preferred internal notation of SYMAP1 by three digit integers: 100,101,...,999. Translations to and from this internal notation make use of several conversion tables, based on the length of the external symbol. Within the computer program Table 1AV contains single character names, such as A, B, X, Y and their three digit integer synonyms. Table 2AV likewise handles two-character names, such as XI, NU, Al, X', etc. Tables 3AV,4AV,...,9AV are provided for names of three or more characters.

The selection of the three digit integers is arbitrary but does affect the lexicographic sequencing of factors within terms and of terms within sums, etc. Thus, if some final result is wanted in ascending powers of Y, say, and in case of ties in ascending powers of X, then the integer for Y should be less than the integer for X and both should be less than the integers of other variables and parameters.

If for instance we represent Y by 501, X by 502, and A, B, C by 601, 602, 603 respectively, then internally Y becomes 501\*\*1.\$ and  $Y^2$  becomes 501\*\*2.\$, etc. The binary representation of the digits and characters within BRIESC 1 and the ordering rules of SYMAP1 then cause algebraic terms in Y, X, A, B, C to be sequenced as follows:

$$\begin{aligned} &Y, YX, YXA, YXB, \dots \\ &Y^2, Y^2X, Y^2XA, \dots, Y^2X^2, Y^2X^2A, \dots \\ &Y^3, Y^3X, Y^3XA, \dots, Y^3X^2, Y^3X^2A, \dots \\ &\dots \\ &X, XA, XAB, XB, \dots \\ &X^2, X^2A, \dots \\ &X^3, X^3A, \dots \\ &\dots \end{aligned}$$

Thus terms with like powers of Y are adjacent, and for a given power of Y those terms with like powers of X are adjacent, etc. Numerical coefficients do not affect this ordering. (Sufficiently high and

negative exponents do affect the ordering, but the adjacency within groups is maintained.) This adjacency property is useful in combining "like" terms after expansions.

If there is a preferred lexicographic order, this should be considered and the entries in Tables LAV, etc, specified appropriately.

Function names, similarly, are represented by integers of either 1 or 2 digits, using the same Tables LAV, etc. For sequencing, however, these integers are considered 10000 times larger at present, in order to keep functions at the right of any term in which they appear.

Since variables and function names are represented by integers, it is necessary that actual constants be distinguished with special marks. In SYMAP1 this is done by appending a dollar sign at the end of each constant. Also each constant is required to contain a decimal point. Thus unity is 1.\$ in canonical notation and zero is .0\$ internally. Decimal fractions at present are carried with up to eight digits if positive or up to seven digits if negative (an arbitrary decision but convenient in a computer with 10 characters per word).

Internally each factor has an explicit exponent, unity if nothing else, and each term has an explicit coefficient. A coefficient or exponent of unity is removed in the reconversion process, however, to keep most displayed expressions as near to normal notation as possible.

Let us consider another example in several variables.

#### IV. EXAMPLE FROM THE THEORY OF CUBIC EQUATIONS

As another example of symbol manipulation in SYMAP1 let us consider the nature of the roots of a cubic equation. Given the polynomial

$$p(x) = x^3 + bx^2 + cx + d ,$$

where b, c, d are real, let us characterize the region of bcd-space where p(x) has three real roots. It is well known that the requirement

for this condition is that the cubic discriminant defined by

$$\Delta(b,c,d) = 18bcd - 4b^3d + b^2c^2 - 4c^3 - 27d^2$$

be non-negative. Let us confirm this relation.

Of the three roots of any cubic with real coefficients, at least one root is real and the other two may be real and equal, real and unequal, or complex conjugates. Therefore, we can let the roots be  $r$ ,  $s+t$ , and  $s-t$  with  $r$  and  $s$  real and  $t$  zero, positive, or a pure imaginary for the three cases.

The real coefficients  $b$ ,  $c$ , and  $d$  are related to  $r$ ,  $s$ , and  $t$  as follows:

$$\begin{aligned} b &= -r - 2s \\ c &= 2rs + s^2 - t^2 \\ d &= -r(s^2 - t^2) \end{aligned}$$

If these last expressions are substituted for  $b$ ,  $c$ , and  $d$  in the discriminant formula, we obtain:

$$\Delta = \Delta_0 + \Delta_1 t^2 + \Delta_2 t^4 + \Delta_3 t^6$$

where the coefficients  $\Delta_j$  are polynomials in  $r$  and  $s$ .

If  $t=0$   $p(x)$  has at least a double root  $s$  and the properties of the discriminant state that  $\Delta$  vanishes. Hence we note that it must follow that the coefficient  $\Delta_0$  vanishes identically:

$$\Delta_0 = 0.$$

This fact can be verified using SYMAP1.

If indeed  $\Delta_0 = 0$ , then our problem of showing

$$\Delta(r,s,t) \geq 0$$

reduces to the equivalent problem of verifying that

$$\Delta/t^2 = \Delta_1 + \Delta_2 t^2 + \Delta_3 t^4,$$

a quadratic in  $t^2$ , is non-negative (when  $r$ ,  $s$ , and  $t$  are real).

Input:

```

B      -1.*R-2.*S
C      2.*R*S+S**2.-T**2.
D      -1.*R*S**2.+R*T**2.
TM2    T** -2.
TM4    T** -4.
IT2    801**2.$
IT4    801**4.$
IZERO  .0$
I4     4.$
I18    18.$
I27    27.$
,,

```

Steps:

Results:

CONVERT	B	IB		b
CONVERT	C	IC		c
CONVERT	D	ID		d
CMULT	ID	ID	ID2	$d^2$
CMULT	IC	IC	IC2	$c^2$
CMULT	IB	IB	IB2	$b^2$
CMULT	IC2	IC	IC3	$c^3$
CMULT	IB2	IB	IB3	$b^3$
CMULT	IB	IC	IBC	bc
CMULT	IBC	ID	IBCD	bcd
CMULT	I18	IBCD	T1	$18bcd = \text{term 1}$
CMULT	IB3	ID	IB3D	$b^3d$
CMULT	I4	IB3D	T2	$4b^3d = \text{term 2}$
CMULT	IB2	IC2	T3	$b^2c^2 = \text{term 3}$
CMULT	I4	IC3	T4	$4c^3 = \text{term 4}$
CMULT	I27	ID2	T5	$27d^2 = \text{term 5}$
CSUBT	T1	T2	S2	
CADD	S2	T3	S3	
CSUBT	S3	T4	S4	
CSUBT	S4	T5	S5	
RECONVERT	S5	DISC		$\Delta$
SPREAD	DISC			(No term without t so $\Delta_0 = 0$ )

Figure 6(a). Generating the Cubic Discriminant

SYMAP1 can be used to determine the literal form of the coefficients  $\Delta_1$ ,  $\Delta_2$ , and  $\Delta_3$  and then the form of the quadratic discriminant

$$E = (\Delta_2)^2 - 4\Delta_1\Delta_3 .$$

For all real  $r$  and  $s$  we would expect to find

$$\Delta_1 \geq 0 \quad \text{and} \quad E \leq 0 .$$

Let us verify these relations also using SYMAP1 as an aid.

Figure 6 indicates a possible set of SYMAP1 steps in support of such verifications. Input forms include  $b$ ,  $c$ ,  $d$  in terms of  $r$ ,  $s$ ,  $t$ , four useful powers of  $t$  (two of them in internal or canonical form), and four constants (also in canonical notation for convenience). The three digit integers 801, 802, and 803 were selected to represent  $t$ ,  $s$ , and  $r$  respectively. Any results are grouped by like powers of  $t$ .

Manipulations include converting  $b$ ,  $c$ , and  $d$  to canonical form, building up  $\Delta$  step by step, displaying  $\Delta$ , isolating the coefficients  $\Delta_1$ ,  $\Delta_2$ , and  $\Delta_3$ , and forming and displaying the quadratic discriminant  $E$ . The observer can draw his conclusions after some additional hand work including some factoring not within the capabilities of SYMAP1.

Figure 7 shows the literal forms of  $\Delta$ ,  $\Delta_1$ ,  $\Delta_2$ ,  $\Delta_3$ , and  $E$  in "spread" form. Note in particular that in  $\Delta$  no term is independent of  $t$ ; so  $\Delta_0 = 0$  as expected.  $\Delta_3$  is found to be the constant 4.  $\Delta_2$  can be factored by hand as  $\Delta_2 = -8(s-r)^2$ , real and non-positive. Similarly  $\Delta_1 = 4(s-r)^4$ , real and non-negative.  $E = \Delta_2^2 - 4\Delta_1\Delta_3 = 0$ , a fact which might not be expected without further analysis, but the desired conclusion  $E \leq 0$  is certainly verified.

Replacing  $\Delta_1$ ,  $\Delta_2$ ,  $\Delta_3$  in  $\Delta/t^2$  leads to:

$$\begin{aligned} \Delta/t^2 &= 4(s-r)^4 - 8(s-r)^2t^2 + 4t^4 \\ \Delta &= 4t^2[(s-r)^4 - 2(s-r)^2t^2 + t^4] \\ &= 4t^2[(s-r)^2 - t^2]^2 \geq 0 \end{aligned}$$

We have thus verified that the discriminant  $\Delta$  is non-negative if the cubic has three real roots. Furthermore,

CONVERT	TM2	ITM2			$t^{-2}$
CONVERT	TM4	ITM4			$t^{-4}$
CMULT	ITM2	S5	S5A		$\Delta \cdot t^{-2}$
RECONVERT	S5A	RS5A			
SPREAD	RS5A				$\Delta_1 + \Delta_2 t^2 + \Delta_3 t^4$
CSBEE	IZERO	IT4	S5A	A2A	$\Delta_1 + \Delta_2 t^2 + \Delta_3 \cdot 0$
CSIME	A2A	A2			$\Delta_1 + \Delta_2 t^2$
CSUBT	S5A	A2	A3		$\Delta_3 t^4$
CMULT	ITM4	A3	IDEL3		$\Delta_3 t^4 \cdot t^{-4} = \Delta_3$
CSBEE	IZERO	IT2	A2	A4A	$\Delta_1 + \Delta_2 \cdot 0$
CSIME	A4A	IDEL1			$\Delta_1 = \Delta_1$
CSUBT	A2	IDEL1	A5		$\Delta_2 t^2$
CMULT	ITM2	A5	IDEL2		$\Delta_2 t^2 \cdot t^{-2} = \Delta_2$
CMULT	IDEL1	IDEL3	T11		$\Delta_1 \Delta_3$
CMULT	I4	T11	T12		$4\Delta_1 \Delta_3$
CMULT	IDEL2	IDEL2	T13		$\Delta_2^2$
CSUBT	T13	T11	IE		$\Delta_2^2 - 4\Delta_1 \Delta_3 = E$
RECONVERT	IDEL1	DEL1			
SPREAD	DEL1				
RECONVERT	IDEL2	DEL2			
SPREAD	DEL2				
RECONVERT	IDEL3	DEL3			
SPREAD	DEL3				
RECONVERT	IE	E			
SPREAD	E				
,,					

Figure 6(b). Analyzing the Cubic Discriminant

-16.*T**2.*S*R**3.	$\Delta(R,S,T)$
+24.*T**2.*S**2.*R**2.	
-16.*T**2.*S**3.*R	
+4.*T**2.*S**4.	
+4.*T**2.*R**4.	
+16.*T**4.*S*R	
-8.*T**4.*S**2.	
-8.*T**4.*R**2.	
+4.*T**6.	
-16.*S*R**3.	$\Delta_1(R,S)$
+24.*S**2.*R**2.	
-16.*S**3.*R	
+4.*S**4.	
+4.*R**4.	
16.*S*R	$\Delta_2(R,S)$
-8.*S**2.	
-8.*R**2.	
4.	$\Delta_3(R,S)$
.0	$E = \Delta_2^2 - 4\Delta_1\Delta_3$

Figure 7. Results of Second Example

$$\begin{aligned}\Delta &= 4t^2[s-r-t]^2[s-r+t]^2 \\ &= 4t^2[(s-t) - r]^2[(s+t) - r]^2 \geq 0.\end{aligned}$$

In this last form it is quite clear that the cubic discriminant  $\Delta$  vanishes if  $t=0$  or if  $r = s+t$  or if  $r = s-t$ , exactly those cases when at least two roots are equal, and  $\Delta$  does not vanish in any other case.

In summary,

$\Delta \geq 0$  for any three real roots  
 $\Delta = 0$  for at least two equal roots (all real)  
 $\Delta > 0$  for three distinct real roots  
 and  $\Delta < 0$  in other cases, namely for one real root and two complex conjugate roots.

SYMAP1 could be used in a similar manner to help derive less well-known relations occurring in original investigations.

## V. SIMPLIFICATION OF FORMS

Several types of simplification occur automatically when SYMAP1 is used. Others can be achieved by appropriate combinations of steps, specified by the user. More flexibility is available when forms are retained in canonical notation; so use of this notation is encouraged.

In general the following simplifications occur, but there are exceptions, especially if complex combinations of steps are carried out without calling for intermediate simplifications. In the formulas below  $a, b, c, n, m, p, q, r$  are real decimal constants (not necessarily integers);  $x, y$  are primitives; and  $t, u, v, w$  are terms similar to the terms of a polynomial.

Within a term like factors are combined, and within a sum like terms are combined. Thus:

$$x^n x^m \rightarrow x^p \quad \text{where } p = n + m$$

$$ax^n y^m + bx^n y^m \rightarrow cx^n y^m \quad \text{where } c = a + b$$

Powers of simple products are expanded:

$$(ax^n y^m)^c \rightarrow bx^p y^q \quad \text{where } b = a^c, p = c \cdot n, q = c \cdot m$$

The distributive law is applied in one direction:

$$ax^c(b_1 x^n + b_2 x^m) \rightarrow r_1 x^p + r_2 x^q$$

but the reverse process (factoring) is not. Special cases of factoring can be done in several steps, however.

Parentheses are removed when possible in special cases, such as:

$$\begin{aligned} t + (u + v - w) &\rightarrow t + u + v - w \\ t - (u + v - w) &\rightarrow t - u - v + w \end{aligned}$$

Simple denominators are replaced:

$$\begin{aligned} u/a &\rightarrow b \cdot u \quad \text{where } b = a^{-1} \\ u/(ax^n) &\rightarrow b \cdot u \cdot x^{-n} \quad \text{where } b = a^{-1} \\ u/(ax^n + by^m) &\rightarrow u \cdot (ax^n + by^m)^{-1} \end{aligned}$$

Special cases of the above rules include:

$$\begin{aligned} x^n \cdot x^{-n} &\rightarrow 1 \\ ax^n - ax^n &\rightarrow 0 \end{aligned}$$

Indeterminate cases must be avoided by the user:

$$0^0 \quad 0/0 \quad 0 \cdot \infty \quad 0^\infty \quad \infty^0 \quad \infty/\infty \quad 1^\infty \quad \infty - \infty$$

Resequencing in some lexicographical order is standard to group similar factors and terms.

## VI. DETAILS OF SPECIFICATION OF MANIPULATIONS

The principal manipulation types were introduced earlier, and examples of their use were given. For those interested additional details are given here.

At present manipulations must be specified in advance on formatted punched cards. (When they become available, on-line keyboards, light

pens, etc. will make SYMAP even more responsive to user needs.) The manipulation type is specified at card column 1 and any additional parameters at columns 11, 21, 31, 41 as needed. The parameters are always names of previously specified forms or variables or of the form currently being generated. Up to 9 letters and digits may be used in names.

Arithmetic operations (add, subtract, multiply, divide, exponentiate) have three parameters: two operands and a result, in the usual order. PRINT and SPREAD require only one parameter, the name of the form to be displayed. CONVERT, RECONVERT, and the simplify types have two parameters each: the given form and a (new) name for the result. (Names cannot be reused with new meanings, even if the old form is not needed further.)

The substitution manipulations have four parameters in the order a, b, c, d, where a represents the form substituted, b the form substituted for, c the form substituted in, and d the result. Thus CSBEE CA CB CC CD means "substitute the form whose name is CA for the form whose name is CB wherever it occurs in the form whose name is CC and store the result with the label CD". Type CSBEE is the most commonly used substitution in SYMAP1. Type SUBSTEE is very similar but uses non-canonical forms.

SUBSTN is merely a renaming operation: "substitute the name a (not the form whose name is a) for the name b wherever b occurs in the form called c and label the result as d". SUBSTNE (name for expression) is similar except that b is the name of the form being substituted for. Finally SUBSTE (expression for name) says to substitute the expression whose name is a for the name b wherever b occurs in c and call the result d. All of these operate on non-canonical forms. Simplification of the results of these operations with SIMEXPR is not always successful, however; so use of canonical forms and CSBEE followed by CSIME is recommended.

The differentiation operations CDIFF and DIFF, for canonical and non-canonical forms, respectively, are currently limited to polynomial-like expressions in the variable of differentiation. Extensions are planned. CDIFF CA Y CB means "differentiate the form whose name is CA with respect to the primitive Y and call the result CB". Since CA is canonical, the variable Y does not appear in CA as such but rather as some previously assigned three digit number such as 101. SYMAP1 looks up the equivalent 101, and differentiates with respect to that symbol, giving a canonical result.

Manipulations are carried out in the order specified by the sequence of cards, and only input and previously completed results can be used in later manipulations. The final operation should be followed by a sentinel card (commas in columns 1 and 2) to terminate manipulations.

## VII. EVALUATION OF EXPRESSIONS

Although the power of SYMAP1 lies in its ability to manipulate polynomials and similar non-numerical expressions, there are occasions when an algebraic form must be evaluated for particular values of its variables. This capability is included in the substitution and simplification operations already described.

Given, say, a polynomial in X and Y where the values  $X = 4$  and  $Y = -3.25$  are to be assigned. The SYMAP1 steps needed include a substitution of 4 for X, a substitution of -3.25 for Y, and a simplification. The substitutions could be reversed if preferred, and simplification could be done after each substitution if display of intermediate results is desired. (See Figure 8.)

Substitution of constants for variables in trigonometric and similar expressions is possible, but simplification is not complete in such cases at present. Future extensions to SYMAP1 may provide for additional evaluations.

Given:

```

X          100
Y          101
PXY       3.*X**.5*Y+X*Y**2.
VX         4.$
VY        -3.25$
,,

```

Steps:

Results:

CONVERT	PXY	CP			P(X,Y)
CSBEE	VX	X	CP	CP1	P(4,Y)
CSIME	CP1	CP2			
CSBEE	VY	Y	CP2	CP3	P(4,-3.25)
CSIME	CP3	CP4			
RECONVERT	CP4	RP			
,,					

Results in SYMAP1 notation:

```

CP          3.$100**.5$,101**1.$+1.$100**1.$,101**2.$
CP1         3.$4.$**.5$,101**1.$+1.$4.$**1.$,101**2.$
CP2         6.$101**1.$+4.$101**2.$
CP3         6.$(-3.25$)**1.$+4.$(-3.25$)**2.$
CP4         22.75$
RP          22.75

```

Figure 8. Evaluations

## VIII. VECTOR OPERATIONS

A limited number of operations on n-tuples of algebraic forms have been provided in SYMAP1. Others could be added. Currently available are VPRINT, VCONVERT, VRECONV, VADD, VSUBTR, and VSIME, which respectively print an n-tuple of consecutive forms, convert an n-tuple into canonical notation, reconvert from canonical form, add two n-tuples, subtract two n-tuples, and simplify two n-tuples.

In each case n, the order of the n-tuple, is specified at card column 41 as a one-digit positive integer. The other parameters (at columns 11, 21, 31 as needed) are those which would be needed for the first form of the n-tuple in non-vector operations.

Thus VPRINT X1 b b 7 would cause printing of the 7-tuple whose elements are the form X1 and its six successors, whatever their names. The various forms need not be similar with respect to length or any other feature. Similarly VCONVERT ETA IETA b 3 would cause ETA to be converted and labeled IETA and would then cause the next two forms after ETA, whatever their names, to be converted and given names much like IETA by adding unity for each in the 9th character of the name. VRECONV acts analogously, as does VSIME.

VADD and VSUBTR, because of their extra parameter, are very slightly different from the above. Thus VADD AB CD EF 4 would add the elements of the 4-tuple whose first element has the name AB to the corresponding elements starting at name CD and give the (unsimplified) results the names EF through EFbbbbbb3, respectively.

These vector operations merely save the user some repetitive card punching when very similar manipulations are needed on several sets of algebraic forms. The user must be careful to keep his sets adjacent, as it is the adjacency and not the similarity of names that matters in vector operations. Incidentally, elements of n-tuples may be manipulated as independent individuals if desired by using their individual names in the usual way. (See Figure 9.)

```

F1      X**2.+3.*X-5.
F2      X**2.-4.*X+2.
F3      X**2.+2.*X+4.
G1      2.*X-4.*Z
G2      3.*X+2.*Y
G3      X-Y+Z
H1      Y**2.-Y
H2      Y+2.*Z
H3      Y-Z
,,

VADD      F1      G1      J1      3
VSUBTR    J1      H1      K1      3
VSIME      K1      SK1      3
VCONVERT   SK1      CK1      3
CMULT      CK1      CK1      CK1A
CMULT      CK1bbbb1 CK1bbbb1 CK1B
CMULT      CK1bbbb2 CK1bbbb1 CK1C
VPRINT     CK1A      3
VRECONV    CK1A      L1      3
,,

```

Figure 9. Vector Operations

## IX. CONCLUSION

The present SYMAP1 has proven to be useful within its realm of application. With it manipulations can be carried out which would be extremely tedious if done manually, and the likelihood of simple errors of arithmetic is reduced. Various alternatives can be tried with little extra effort. Thus a researcher can be freed for more challenging investigations and more quickly be shown the effects of changes of attack on a given problem. Potential users of SYMAP1 are encouraged to try it.

Additional features will be added to the program as they are developed. Several of these are planned for the near future, and still others are currently being studied. Potential users of SYMAP1 on BRLESC 1 are encouraged to discuss their needs with the author.

#### REFERENCES

1. L. W. Campbell and G. A. Beck, "The FORAST Programming Language for ORDVAC and BRLESC (Revised)," Ballistic Research Laboratories Report No. 1273, March 1965.
2. L. W. Campbell and G. A. Beck, "BRLESC I/II FORTRAN," Aberdeen Research and Development Center Technical Report No. 5, March 1970.

# DISTRIBUTION LIST

<u>No. of</u> <u>Copies</u>	<u>Organization</u>	<u>No. of</u> <u>Copies</u>	<u>Organization</u>
20	Commander Defense Documentation Center ATTN: TIPCR Cameron Station Alexandria, Virginia 22314	1	Commanding General U. S. Army Electronics Command ATTN: AMSEL-DL Fort Monmouth, New Jersey 07703
1	Director Advanced Research Projects Agency ATTN: Dr. S. Lukasik Department of Defense Washington, D. C. 20301	1	Commanding General U. S. Army Missile Command ATTN: AMSMI-R Redstone Arsenal, Alabama 35809
1	Commanding General U. S. Army Materiel Command ATTN: AMCDL Washington, D. C. 20315	1	Commanding General U. S. Army Tank-Automotive Command ATTN: AMSTA-RHFL Warren, Michigan 48090
1	Commanding General U. S. Army Materiel Command ATTN: AMCRD,Dr.J.V.R.Kaufman Washington, D. C. 20315	2	Commanding Officer U. S. Army Mobility Equipment Research & Development Center ATTN: Tech Docu Ctr,Bldg 315 AMSME-RZT Fort Belvoir, Virginia 22060
1	Commanding General U. S. Army Materiel Command ATTN: AMCRD-TE Washington, D. C. 20315	1	Commanding General U. S. Army Munitions Command ATTN: AMSMU-RE Dover, New Jersey 07801
1	Commanding General U. S. Army Materiel Command ATTN: AMCRD-TP Washington, D. C. 20315	1	Commanding Officer U. S. Army Edgewood Arsenal ATTN: SMUEA-TSTD Edgewood Arsenal, Maryland 21010
1	Commanding General U. S. Army Aviation Systems Command ATTN: AMSAV-E 12th & Spruce Streets St. Louis Missouri 63166	1	Commanding Officer U. S. Army Frankford Arsenal ATTN: SMUFA-L1000,S.Eisman Philadelphia, Pennsylvania 19137

# DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>	<u>No. of Copies</u>	<u>Organization</u>
2	Commanding Officer U. S. Army Picatinny Arsenal ATTN: SMUPA-V, Mr. J. Pearson SMUPA-D, Dr. S. Bernstein Dover, New Jersey 07801	1	Commanding Officer U. S. Army Research Office (Durham) ATTN: Dr. A. Galbraith Box CM, Duke Station Durham, North Carolina 27706
1	Commanding General U. S. Army Weapons Command ATTN: AMSWE-RE Rock Island, Illinois 61202	2	Director U. S. Army Research Office ATTN: Dr. I. Hershner Mr. M. Weik, Jr. 3045 Columbia Pike Arlington, Virginia 22204
1	Director U. S. Army Advanced Materiel Concepts Agency Washington, D. C. 20315	1	Mathematics Research Center U. S. Army University of Wisconsin ATTN: Dr. J. Rosser Madison, Wisconsin 53706
1	Director U. S. Army Aeronautical Research Laboratory Moffett Naval Air Station California 94035	3	Commander U. S. Naval Air Systems Command ATTN: AIR-604 Washington, D. C. 20360
1	Commanding Officer U. S. Army Harry Diamond Laboratories ATTN: AMXDO-TD/002 Washington, D. C. 20438	3	Commander U. S. Naval Ordnance Systems Command ATTN: ORD-9132 Washington, D. C. 20360
1	Commanding Officer U. S. Army Materials & Mechanics Research Center ATTN: AMXMR-ATL Watertown, Massachusetts 02172	1	Commanding Office & Director U. S. Naval Ship Research & Development Center ATTN: Miss S. Voight Washington, D. C. 20007
1	Commanding General U. S. Army Natick Laboratories ATTN: AMXRE, Dr. D. Sieling Natick, Massachusetts 01762	1	Director U. S. Naval Research Laboratory ATTN: Code 2020 Washington, D. C. 20390
1	Office of the Vice Chief of Staff ATTN: CSAVCS-W-TIS Department of the Army Washington, D. C. 20310		<u>Aberdeen Proving Ground</u> Ch, Tech Lib Marine Corps Ln Ofc CDC Ln Ofc

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D		
(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)		
1. ORIGINATING ACTIVITY (Corporate author) U.S. Army Aberdeen Research and Development Center Ballistic Research Laboratories Aberdeen Proving Ground, Maryland		2a. REPORT SECURITY CLASSIFICATION Unclassified
		2b. GROUP
3. REPORT TITLE SYMAP1 - AN EXPERIMENTAL SYMBOL MANIPULATION PROGRAM		
4. DESCRIPTIVE NOTES (Type of report and inclusive dates)		
5. AUTHOR(S) (First name, middle initial, last name) George C. Francis		
6. REPORT DATE August 1970	7a. TOTAL NO. OF PAGES 34	7b. NO. OF REFS 2
8a. CONTRACT OR GRANT NO.  b. PROJECT NO. 1T061102A14B  c.  d.		9a. ORIGINATOR'S REPORT NUMBER(S) BRL Memorandum Report No. 2060  9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
10. DISTRIBUTION STATEMENT This document has been approved for public release and sale; its distribution is unlimited.		
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY U.S. Army Materiel Command Washington, D.C.
13. ABSTRACT SYMAP1 is a BRLESC computer program designed to carry out a variety of algebraic symbol manipulations including arithmetic operations, substitution, certain kinds of simplifications, and rudimentary differentiation. Main emphasis has been on polynomials in several variables (including truncated power series), but certain other mathematical forms can also be manipulated. Examples of certain applications to numerical analysis and theory of equations are included.		

DD FORM 1473

REPLACES DD FORM 1473, 1 JAN 64, WHICH IS OBSOLETE FOR ARMY USE.

Unclassified

Security Classification

Unclassified

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Symbol manipulation Algebraic forms Substitution in forms Simplification of forms Symbolic differentiation Vector operations BRLESC computer program Numerical analysis example Roots of cubics example						

Unclassified

Security Classification